RZ 3354 (# 93400) 05/22/01
Computer Science 17 pages

Research Report

The Selfish Node: Increasing Routing Security for Mobile Ad Hoc Networks

Sonja Buchegger[1] and Jean-Yves Le Boudec[2]

[1]IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
sob@zurich.ibm.com

[2]EPFL-DSC
1015 Lausanne
Switzerland
jean-yves.leboudec@epfl.ch

IBM Research
Almaden · Austin · Beijing · Delhi · Haifa · T.J. Watson · Tokyo · Zurich

The Selfish Node: Increasing Routing Security for Mobile Ad Hoc Networks

Sonja Buchegger[1] and Jean-Yves Le Boudec[2]

[1] *IBM Research, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland*

[2] *EPFL-DSC, 1015 Lausanne, Switzerland*

Abstract

Nodes in mobile ad hoc networks do not rely on a central infrastructure but relay packets originated by other nodes. Mobile ad hoc networks can work properly only if the participating nodes collaborate in routing and forwarding. For individual nodes it might be advantageous not to collaborate, though. The new routing protocol extensions presented in this paper make it possible to detect and isolate misbehaving nodes, thus making it unattractive to deny collaboration. In the presented scheme, trust relationships and routing decisions are made based on experienced, observed, or reported routing and forwarding behavior of other nodes. A hybrid scheme of selective altruism and utilitarism is presented to strengthen mobile ad hoc network protocols in their resistance to security attacks, while aiming at keeping network throughput high. This paper focuses particularly on the network layer, using the Dynamic Source Routing (DSR) protocol as an example.

1 Introduction

Mobile ad hoc networks do not rely on any fixed infrastructure but communicate in a self-organized way. Their security requirements cannot be addressed in the same way as those of infrastructure-based or wired networks, because mobile ad hoc networks are vulnerable to attacks unknown in these traditional networks. An example of a mobile ad hoc network is being developed within the Terminodes[1] project [9], so called because the devices act as nodes and terminals at the same time, and forward packets that are destined to other nodes. The Terminodes project is about large mobile ad hoc networks. It is different from other mobile ad hoc networks as proposed in the MANET (mobile ad hoc networks) working group of the IETF [10] in that the network is a wide-area, self-organized network. The wide area aspect raises specific scalability issues for the number of nodes and the distance between communicating nodes in terms of both physical distance and number of intermediate nodes. Furthermore, the Terminodes network is not limited to an organization who could enforce collaboration. Therefore, there is a need for incentives to collaborate in order to encourage the nodes to forward packets, although doing so consumes their resources.

The issues discussed in this paper are relevant for both MANET-style and Terminodes networks, however, the need for efficient solutions and protocols is stronger in the Terminodes network. This stronger need is due to the increased scalability and distribution requirements, as well as an "open world assumption" that the participants of a Terminodes network are most likely not of the same organization. The focus of the MANET working group is on routing protocols, and relatively little has been published concerning the security of these protocols [7].

One of the protocols presented and discussed in the MANET working group of the IETF is the Dynamic Source Routing (DSR) protocol [11]. The protocol is briefly presented in this paper and serves as an example of security vulnerabilities and what can be done to eliminate them. An extension to DSR is proposed for this purpose.

The remainder of this paper is organized as follows: After Section 2 on particular security issues in mobile ad hoc networks and related work in Section 3, a new approach to security and collaboration is motivated in Section 4 and outlined as a protocol in Section 5. Assumptions made in order that this new protocol works are explained in Section 6, followed by a description of the methodology of the approach and simulation in Section 7. The rest of this paper consists of an outline of future work in Section 9 and the concluding Section 10.

2 Additional Security Issues for Mobile Ad Hoc Networks

In addition to authentication, integrity, confidentiality, availablity, access control and non-repudiation (see [18] for details), mobile ad hoc networks raise the following security issues:

Collaboration and fairness: Although not an issue in infrastructure-based networks, there has to be an incentive for a node to forward messages that are not destined to itself. Nodes are assumed to be greedy, selfish, and economic. Attacks include incentive mechanism exploitation by message interception, copying, or forging; incorrect forwarding; and

[1] http://www.terminodes.org

bogus routing advertisement. There is a trade-off between good citizenship and resource consumption, so nodes have to economize on their resources. At the same time, however, if they do not forward messages, others might not forward either, thereby denying them service. Total non-collaboration with other nodes and only exploiting their readiness to collaborate is one of several boycotting behavior patterns. Nodes could decide to not collaborate with normal well-behaved nodes, exploit their collaboration, and then restrict access to their own resources to other colluding nodes, thus forming a parallel 'underground' network. In such a network the selfish nodes would deprive the normal nodes of resources and at the same time exploit the resources of the normal nodes.

Confidentiality of location: In some scenarios, for instance in a military application, routing information can be equally or even more important than the message content itself [6]. It can be necessary to protect the privacy of routing information as well as not to reveal the whereabouts of a given node. This, however, prevents the use of routing information by intermediate nodes or neighbors even for security purposes. Related to the confidentiality of location is also the traceability of nodes, both a physical location and the tracking down of a node identity based on its routing traffic.

No traffic diversion: Routes should be advertised and set up adhering to the chosen routing protocol and should truthfully reflect the knowledge of the topology of the network. By diverting the traffic in the following ways, nodes can work against that requirement:

o **Routing:** To get information necessary for successful malicious behavior, nodes can attract traffic to themselves or their colluding nodes by means of false routing advertisements. There are many ways to make up a bogus route that exhibits the properties of a good route and is subsequently preferred over real routes (see Section 7 for details). These bogus routes can be made to stay longer in routing caches. In order not to raise suspicion, malicious nodes can keep a copy of the received messages and actually forward the messages to the originally intended destination. Although only suitable for devices that have enough power, a lot of information can be gathered this way by malicious nodes for later use to enable more sophisticated attacks.

  Denial-of-service attacks can be achieved by bogus routing information (injecting of incorrect routing information or replay of old routing information or 'black hole routes') or by distorting routing information to partition the network or to load the network excessively, thus causing retransmissions.

o **Forwarding:** Nodes can decide to forward messages to partners in collusion for analysis, disclosure, or monetary benefits, or may decide not to forward messages at all, thus boycotting communications.

Motivation for attacks: The lack of infrastructure and organizational environment of mobile ad hoc networks offer special opportunities to attackers. Without proper security, it is possible to gain various advantages by malicious behavior: better service than cooperating nodes, monetary benefits by exploiting incentive measures or trading confidential information; saving power by selfish behavior, preventing someone else from getting proper service, extracting data to get confidential information, and so on. In contrast, section 4 provides a rationale why collaboration can pay off.

3 Related Work

Anderson and Stajano [1] authenticate users by 'imprinting' according to the analogy of ducklings acknowledging the first moving subject they see as their mother, but enabling the devices to be imprinted several times. Haas employs threshold security, allowing for several corrupted nodes or collusions [7]. Garcia-Luna-Aceves et al. [17] looked at security of distance vector protocols in general.

For the Terminodes project, incentives to collaborate by means of so-called nuglets [3] that serve as a per-hop payment in every packet have been suggested by Buttyan et al. to ensure forwarding. The scheme suggested here in the following sections addresses additional issues in the network layer such as traffic diversion.

Marti et al. [13] observed increased throughput in mobile ad hoc networks by complementing DSR with a watchdog (for detection of malicious behavior) and a 'pathrater' (for trust management and routing policy, every used path is rated), which enable nodes to avoid malicious nodes in their routes. Their approach raises scalability issues, because everyone keeps a rating about every other node, which is not suitable for Terminodes networks and an open world assumption, unless the rating times out within a suitable time or nodes only keep ratings about their neighbors. Nor does their approach punish malicious nodes that do not collaborate, but rather relieves them of the burden of forwarding for others, whereas their messages are forwarded without complaint. This way, the malicious nodes are rewarded and reinforced in their behavior. Although this increases the total network throughput, it is undesirable. We would like to achieve the contrary, namely that malicious behavior and non-collaboration are punished and do not pay off. Detection of this kind of behavior is key but not the only point. The detection has to lead to a reaction of other nodes such that it results in a disadvantage for the malicious node. This punishment can very well be by means of isolation, but not positive isolation in being isolated from the society's duties but above all the society's rights. Packets of malicious nodes should, upon detection of the node being malicious, not be forwarded by the normally behaving nodes. If, however, a node was wrongly accused of being malicious or turns out to be a repenting criminal equivalent who is no longer malicious and has behaved normally for a certain amount of time, some sort of 're-socialization' and re-integration into the network communications should be possible.

Prevention, detection and reaction: According to Schneier [16], a prevention-only strategy only works if the prevention mechanisms are perfect; otherwise, someone will find out how to get around them. Most of the attacks and vulnerabilities have been the result of bypassing prevention mechanisms. Given this reality, detection and response are essential.

4 The Selfish Node

4.1 The Selfish Gene

As explained in [4], reciprocal altruism is beneficial for every biological system when favors are granted simultaneously, so there is an intrinsic motivation for cooperation due to instant gratification. The benefit of behaving well is not so obvious in the case of a delay between granting a favor and repayment, which is the case when, in mobile ad hoc networks, nodes forward for each other. A biological example used in [4] explains the survival chances (and thus gene selection) of birds grooming parasites off each other's head, which they cannot clean themselves. Dawkins divides birds into two types: ‘suckers’ which always help and ‘cheats’ which have other birds groom parasites off their head but fail to return the favor. In this system, clearly the cheats have an advantage over the suckers, but both are driven to extinction over time. Dawkins then introduces a third kind of bird, the ‘grudger’ which starts out being helpful to every bird, but bears a grudge against those birds that do not return the favor and subsequently no longer grooms their head. According to Dawkins, simulation has shown that when starting with a majority population of cheats and marginal groups of both suckers and grudgers, the grudgers win over time. Winning is defined as having the greatest benefit, assuming a cost for grooming another bird’s head and a profit of having one’s head groomed, a loss leading to extinction and profit leading to multiplication of the species. The rationale is as follows: The suckers help more than they get favors due to the large number of cheats, so the number of suckers decreases, while the number of cheats increases. The grudgers also suffer from some loss, but less than the suckers. Once the suckers are extinct, the grudgers grow rapidly at the expense of the cheats, because they don’t help a cheat twice and cheats are also not helped by other cheats. After a while, the number of cheats decreases more slowly, because the probability of a first-help by a grudger increases with a higher population of grudgers. Over all, the population of the grudgers grows, whereas the other species become extinct.

4.2 Application and Improvements

Defining suitable cost and profit to routing and forwarding favors and keeping a history of experiences with non-collaborating nodes achieves the same as the grudger species, driving the cheats out of business. In a very large ad hoc network, convergence can be very slow, and keeping a history of all bad experiences with other nodes equals large storage requirements and long lists to go through. Therefore, we suggest the following ideas, which are incorporated in a protocol explained in the next section, to speed up the winning of grudger nodes. The suggestions also take the resulting throughput of the network into consideration:

- employ ‘neighborhood watch’ to be warned by watching what happens to other nodes in the neighborhood, before nodes have to make a bad experience themselves,
- share information of experienced malicious behavior with friends and learn from them.

5 The Protocol

The protocol containing the improvements to the grudger’s scheme consists of the following components as shown in Figure 1:

The Monitor

The Reputation System

The Path Manager

The Trust Manager

The components are present in every node and they are described in detail subsequently:

5.1 Components

5.1.1 The Monitor (Neighborhood Watch)

In a networking environment, the nodes most likely to detect non-compliant 'criminal' behavior are the nodes in the vicinity of the criminal and in some cases the source and the destination, if they detect unusual behavior or do not get proper responses. The latter is not always the case, for instance in the case of replay. One approach to protocol enforcement and detection of damaging behavior (intrusion, misuse of collaboration incentives, denial of service, etc.) suggested here is the equivalent of a 'neighborhood watch', where nodes locally look for deviating nodes.

When deviant behavior is detected, punishment (behavioral conditioning), is also carried out by the neighboring nodes. Each node can act upon its own observations and optionally upon warnings received from other trusted nodes. To spread the news further, the nodes can tell the nodes they trust and want to protect.

The neighbors of the neighborhood watch can detect deviances by the next node on the source route by either listening to the transmission of the next node or by observing route protocol behavior. By keeping a copy of a packet while listening to the transmission of the next node, any content change can also be detected. In general, the following types of misbehavior can be indicated:

- no forwarding (of control messages nor data),
- unusual traffic attraction (advertises many very good routes or advertises routes very fast, so they are deemed good routes),
- route salvaging (i.e. rerouting to avoid a broken link), although no error has been observed,
- lack of error messages, although an error has been observed,
- unusually frequent route updates,
- silent route change (tampering with the message header of either control or data packets).

For the deviating behavior listed above, reasonable thresholds have to be introduced that may not be exceeded by the supposedly malicious node. Two types of neighbors exist: the preceding node in the source route, and any other node one hop away from the observed node. These two types of neighbors have different capabilities. The neighbor node that is on the same path as the observed node has additional route information and can detect whether the packet was forwarded to the next hop in the route, whereas the routing protocol related behavior can be observed by any neighbor within a one-hop radius.

As a component within each node, the monitor registers deviations of normal behavior and manages them in the watch table. The watch table contains a list of events, thresholds, counters, and timers. As soon as a given bad behavior occurs more often than a configurable threshold, an ALARM message is triggered and sent to the reputation system of the observing node itself as well as to its trust manager in order to be potentially sent to friends as a warning (see Fig. 1). Nodes can decide on which types of events to include in the watch table and how to define the threshholds and timers according to their needs.

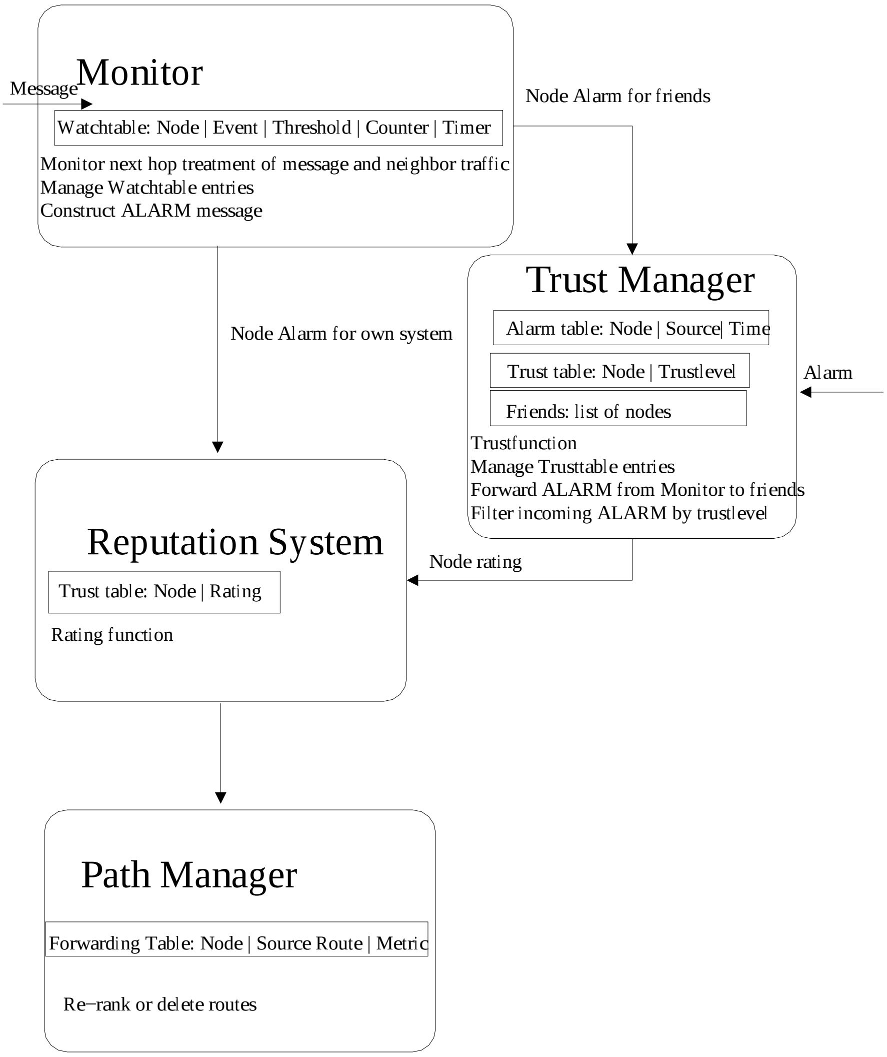
Monitor
Message
Watchtable: Node | Event | Threshold | Counter | Timer
Monitor next hop treatment of message and neighbor traffic
Manage Watchtable entries
Construct ALARM message
Node Alarm for friends
Node Alarm for own system
Trust Manager
Alarm table: Node | Source| Time
Trust table: Node | Trustlevel
Friends: list of nodes
Alarm
Trustfunction
Manage Trusttable entries
Forward ALARM from Monitor to friends
Filter incoming ALARM by trustlevel
Reputation System
Trust table: Node | Rating
Rating function
Node rating
Path Manager
Forwarding Table: Node | Source Route | Metric
Re−rank or delete routes

Figure 1: **Trust architecture within each node:** The frames of each component contain variables stored in tables; below the frames, the functions provided by the component are listed; and the arrows symbolizes exchanged messages

If a node detects that a suspected malicious node is on the source route, it can inform the source about it. It should not pay off *per se* to denounce someone, otherwise allegations would arise unnecessarily and cause a climate of distrust and Spanish Inquisition-like suspicion. In the long run, however, valid accusations help to keep the network functioning by enabling isolation of malicious nodes, thus increasing throughput.

5.1.2 The Trust Manager

In an ad hoc environment, trust management has to be distributed and adaptive [2]. This component deals with incoming ALARM messages originated both from the node's own monitor as well as from outside friends. The trust manager administrates a table of friends and how much they are trusted. This is similar to trust management in PGP (see below). Incoming ALARM messages are filtered according to the trust level of the reporting node. The trust manager consists of the following components:

- Alarm table containing information about received alarms,
- Trust table managing trust levels for nodes,
- Friends list containing all friends a node sends alarms to.

These functions are performed by the trust manager:

- Trust function to calculate trust levels,
- Trust table entries management for trust level administration,
- Forwarding of ALARM messages,
- Filtering of incoming ALARM messages according to the trust level of the reporting node.

The term 'friend' subsequently refers to a trusted node. Trust is important when making a decision about the following issues:

- providing or accepting routing information,
- accepting a node as part of a route,
- taking part in a route originated by some other node.

Several rationales go into designing the components of the trust manager:

PGP In PGP [20], several levels of trust can be expressed, e.g. 'unknown', 'none', 'marginal', and 'complete'. These levels of trust are used when certifying public keys. When PGP is calculating the validity of a public key, it examines the trust level of all the attached certifying signatures. It computes a weighted score of validity. For example, two marginally trusted signatures might be deemed credible as one completely trusted signature. The weighting scheme is adjustable to require a different number of marginally trusted signatures to judge a key as valid.

A similar mechanism is used here for mobile ad hoc networks not only for key validation and certification, but for trust management for routing and forwarding as done in the trust manager.

Rumor spreading Different ways of rumor or gossip spreading and their similarities to epidemics have been investigated by Demers et al. [5]. The ways of rumor spreading can be adapted for information flow to friends. Although not included in the protocol now, received ALARM messages could be forwarded to friends in the way rumors are spread. One way of achieving this effectively is presented in [5] and is called 'rumor mongering'.

Transitive trust in a small world According to the 'small world phenomenon' (first coined by Milgram in the 1960s) [12] everyone is connected to any other person by only a small number of acquaintances, a theory that was later also called 'six degrees of separation'. In a mobile ad hoc environment, this means that relations such as neighbor (a,b) or friend (a,b) can be daisy-chained to find a surprisingly short link between two given nodes. This phenomenon can be exploited for trust management by giving priority to a node if it has fewer degrees of separation in terms of the friend relationship than another node providing routing information to the same destination. Trust can be defined as a function of separation and previous experience. A PGP-like trust management with algorithms for trust transitivity can be used for the distributed environment. Paired with the small world phenomenon, short trust chains can be formed and applied. Although the relationships can be defined easily, it is difficult to find out the shortest link between two nodes especially in a distributed system. The problem is similar to the routing problem itself in large mobile ad hoc networks. Centralized systems already exist to trace the link chain between two people [8]. The small world phenomenon only works in graphs or networks with a certain topology: highly clustered networks with few links between the clusters. It remains to be seen whether this holds true for mobile ad hoc networks.

5.1.3 The Reputation System (Node Rating)

Reputation systems are used in some online auctioning systems. They provide a means of obtaining a quality rating of participants of transactions by having both the buyer and the seller give each other feedback on how their activities were perceived and evaluated. With these auctioning systems, transaction partners can then be rated according to the number of transactions already completed as well as the grades obtained from their former buyers or sellers. There are different representations of the ratings sporting either an average value of the rating, or all obtained ratings or the latest ratings up to a specific time. The latter enables 'bad' trading partners to have their rating timed out and be improved by consistent 'good' behavior over the specified period of time, i.e. they are not punished forever for having shown bad behavior in the past. Such rating schemes enforce a preference of good trading partners over bad ones, thus isolating the bad or unreliable ones from the business. In the networking world, this would mean, that bad nodes would be isolated from communications within the network. The auctioning analogy, however, cannot be applied directly to a mobile ad hoc network context, since the ratings are stored on one or more central auction servers, an infrastructure that is not available in ad hoc networks. Therefore, in order to apply such a rating scheme, it has to work in a distributed fashion, which raises the usual centralized versus distributed approach questions such as additional overhead, consistency, redundancy handling, and so forth. These questions will be addressed in a simulation of the protocol that is described in this paper. Similar to the auctioning feedback are some consumer or opinion sites, where comments on experiences with products and evaluations are entered. In this version, no transaction has to forego an evaluation and rating, which makes it easier to give early warnings but also renders them less credible. For a detailed explanation of reputation systems see [15].

In order to avoid centralized rating, local rating lists and/or black lists are maintained at each node and potentially exchanged with friends. The nodes can include black sheep in the route request to be avoided for routing, which also alarms nodes on the way. Nodes can look up senders in the black list containing the nodes with bad rating before forwarding anything for them. The problem of how to distinguish alleged from proven malicious nodes and thus how to avoid false accusations can be lessened by timeout and subsequent recovery or revocation lists of nodes that have behaved well for a specified period of time. Another problem is scalability and how to avoid blown-up lists, which can also be addressed by timeouts.

The reputation system used in this protocol manages a table consisting of entries for nodes and their rating. Whenever an ALARM message comes into the reputation system component, the entry of the node contained in the message is changed according to a rate function that assigns different weights to the ALARM depending on the source of the ALARM:

- Own experience: greatest weight,
- Observations: smaller weight,
- Reported experience: weight function according to PGP trust.

Once the weight of the ALARM has been determined, the entry of the node that misbehaved is changed accordingly. The questions of positive change and timeout are still to be addressed in detail.

5.1.4 The Path Manager

The path manager performs the following functions:

- Path re-ranking according to security metric,
- Path deletion of path containing malicious nodes,
- Action on receiving request for a route from a malicious node,
- Action on receiving request for a route a malicious node in the source route.

5.2 The ALARM Message

In order to convey the warning information, an ALARM message is sent. This message contains the type of protocol violation, the number of occurrences observed, whether the message was self-originated by the sender, the address of the reporting node, the address of the observed node and the destination address (either the source of the route or the address of a friend that might be interested).

5.3 Information Flow

The suggested scheme works as an extension to a routing protocol. In this example, normal DSR information flow (ROUTE REQUEST, ROUTE REPLY messages) as explained in Section 7 takes place. Once non-cooperative behavior has been detected and exceeds threshold values, an ALARM message is sent. Figures 2 through 5 show the flow of messages and data from route discovery to the detection of malicious behavior and subsequent rerouting. In more detail:

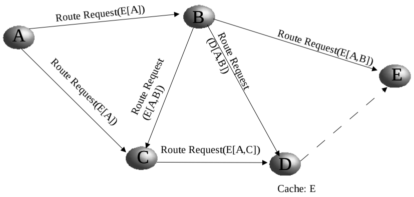

Figure 2: Route request: *A wants to send to E.*

Figure 2 shows DSR route discovery for a path from node A to node E. Every node forwards the request to its neighbors unless it has already received the same route request or has a path cache entry for the desired destination.

Figure 3 shows the reply messages of the destination node itself, node E, and from node D, which has a path to E. The reply message contains the reversed source route to the destination and is sent to the source. In the case of unidirectional links, or if generally the route can not be reversed, node E would send the reply along a path to A that it has in its route cache. If there is no path to A in the route cache, E has to perform a route discovery itself to get to A. In this route request, the already found path from A to E is included.

In Figure 4 data flow is from node A to node E via node C and D. In this case, node A has chosen this route according to some metrics and preferred it over the route via B. During the data flow, node C detects that node D does not behave correctly. In this example, node

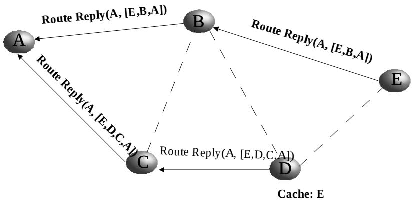

Figure 3: Route reply: *both D and E know a path to E.*

D does not forward the data destined for node E. After the occurrence of the bad behavior of node D was observed by node C for a number exceeding a threshold, node C triggers an ALARM message to be sent to the source, node A.

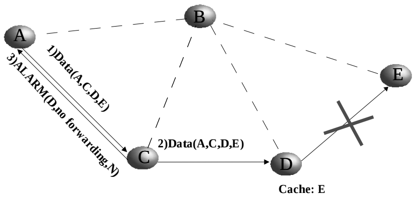

Figure 4: Data flow and alarm: *A sends data and receives an ALARM from C that D does not forward.*

Upon reception of the ALARM message as shown in Figure 5, node A acknowledges the message to the reporting node C and decides to use the alternate path via node B to send the data to the destination node E.

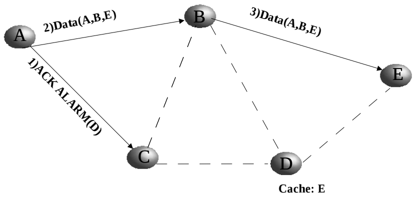

Figure 5: Act on alarm: reroute: *A uses an alternate path to E.*

6 Assumptions

Truly Observable Behavior It is assumed to be possible to listen in on the communications of the neighbors. This means that nodes keep a copy of the packet they send on to another node and listen to the traffic sent on by that node. After they successfully overhear the packet being sent on by the next node, they clear it from their own memory. The types of behavior judged bad or malicious in the monitor component of the protocol presented in this paper are restricted to those that are really observable, given that the previously mentioned assumption holds. The ability to overhear communications in the neighborhood is also assumed by certain MANET routing protocols such as DSR, which has the option of 'passive acknowledgement'.

Authentication is a prerequisite for the protocol to work and assumed to exist here. One way to achieve authentication is by using PGP along with distributed certification authorities. Without authentication, nodes can denounce each other at will and a trust management scheme is not feasible.

The assumption about the behavior being observable has to be complimented by authentication for a node to know whether what it observed was really done by who it thinks it was done. Therefore, there has to be local authentication with nodes in the neighborhood. In addition, authentication with friends that can be in remote places is needed to know that a message was really sent by a friend. For this purpose, PGP can be used, since the number of friends will be substantially smaller than the number of nodes in the network, so the number of keys will be small enough to be stored in each node.

To determine identities, certification authorities are used. In this mobile ad hoc environment, however, a decentralized trust management is called for, so there will have to be at least several distributed authorities throughout the mobile ad hoc network if PGP is to be used as the general authentication mechanism. Alternative solutions could be based on the following: Authentication starting upon entrance into the network and lasting throughout the participating time has to be done in a distributed self-organized manner. The concept of friends can be used, achieving a higher degree of trust by transitivity, when a node is authenticated or introduced by a trusted device analogous to certificate authorities and trusted third parties. In order to detect that a formerly trusted device has been compromised, threshold security mechanisms requiring consensus or secret sharing of a number of nodes can be used (see [7] for an example). If Byzantine robustness [14] is achieved, a network can still function properly in the presence several malfunctioning nodes if there are enough nodes that work normally.

7 Methodology

For the simulation of the protocol presented the following methodology has been chosen:

7.1 Means: DSR, GloMoSim

GloMoSim [19] is used to implement the protocol extensions presented in this paper. DSR has already been implemented in GloMoSim and is modified accordingly:

7.1.1 Routing Example: DSR

Dynamic Source Routing is a protocol developed for routing in mobile ad hoc networks and was proposed for MANET by Broch, Johnson and Maltz at Carnegie Mellon University [11]. In a nutshell, it works as follows: Nodes send out a ROUTE REQUEST message, all nodes that receive this message forward it to their neighbors and put themselves into the source route unless they have received the same request before. If a receiving node is the destination, or has a route to the destination, it does not forward the request, but sends a REPLY message containing the full source route. It may send that reply along the source router in reverse order or issue a ROUTE REQUEST including the route to get back to the source, if the former is not possible due to asymmetric links. ROUTE REPLY messages can be triggered by ROUTE REQUEST messages or gratuitous. After receiving one or several routes, the source picks the best (by default the shortest), stores it, and sends messages along that path. In general, the better the route metrics (number of hops, delay, bandwidth or other criteria) and the sooner the REPLY arrived at the source (indication of a short path - the nodes are required to wait a time corresponding to the length of the route they can advertise before sending it in order to avoid a storm of replies), the higher the preference given to the route and the longer it will stay in the cache. In case of a link failure, the node that can not forward the packet to the next node sends an error message towards the source. Routes that contain a failed link, can 'salvage' the route by bypassing the bad link.

7.1.2 Attacking DSR

We found the following ways of attacking DSR, targeting availability, integrity, confidentiality, non-repudiation, authentication, access control or any combination thereof:

- Incorrect forwarding: acknowledge ROUTE REQUEST, send new request or do not forward at all. This works only until upper layers find out.
- Bogus routing information or traffic attraction: reply to ROUTE REQUEST, also gratuitous, to advertise a non-existent or wrong route.
- Salvage a route that is not broken. If the salvage bit is not set, it will look like the source is still the original one.
- Choose a very short reply time, so the route will be prioritized and stay in the cache longer.
- Set good metrics of bogus routes for priority and remaining time in the cache.
- Manipulate flow metrics for the same reason.
- Do not send error messages in order to prevent other nodes from looking for alternative routes.
- Use bogus routes to attract traffic to intercept packets and gather information.
- Use promiscuous mode to listen in on traffic destined for another node.
- Cause a denial-of-service attack caused by overload by sending route updates at short intervals.

7.2 Steps

The approach of this simulation is to

- place N mobile nodes using DSR on a plane,
- select M nodes to be malicious,
- for each node select F friends,

o generate traffic between nodes while moving,

o gather statistics on throughput/overhead/etc.,

o vary parameters.

7.3 Metrics

In particular, the following issues are of interest:

o Throughput

x in a normal well-behaved network,
x with some malicious nodes but no defence,
x with some malicious nodes and reaction to bad experience,
x and local monitoring,
x and warning of friends.

o Overhead/control messages

o Cost/benefit analysis

x for individual nodes,
x over the entire network.

o Scalability of the protocol

o Relationship between:

x number of nodes in the network,
x number of malicious nodes,
x number of friends per node.

We are interested in the effects on throughput, overhead, etc. by varying the following parameters:

o number of nodes,

o number of malicious nodes,

o number of friends,

o distribution of the above,

o cost/benefit function,

o mobility model,

o speed,

o routing protocol.

8 Results

The simulation has shown that even if the DSR protocol is only fortified by monitoring forwarding and reacting according to the protocol introduced in this paper, only up the first few packets are dropped (according to the defined threshold plus the time it takes to react) in the fortified version of DSR, whereas all of the packets are dropped in the case of malicious intermediate nodes but without defense of the remaining cooperating nodes.

The more applications between different nodes take place, the higher the synergies they create in terms of reacting on malicious nodes and even fewer packets are dropped. This can be explained easily by the fact that nodes can promisuously overhear ALARM messages and then do not insert paths containing reportedly malicious nodes in their path cache. When they then start an application like FTP or Telnet, they do not have paths containing malicious nodes in their cache and hence do not need to reroute around the malicious paths.

9 Future Work - Next Steps

The next steps will consist of implementing more of the approaches discussed so far in simulations for evaluation, and issues that have not been addressed in this paper, for instance what happens to a node in a remote location, where friends might be far away, or how to deal with colluding nodes. Another aspect to investigate is how groups of nodes only collaborating with 'club members' do in the long run, and whether it would be possible to have coexisting interest groups that do not cooperate outside of the group. Also, there is an open question of why nodes would warn friends, which again has no immediate gratification but incurs the instant cost of sending messages. A meta-grudgers scheme, direct reimbursement for warning effort by an amount equal to the expense incurred for alarming or taking round-robin shifts in neighborhood-watch scheme are possible directions for research.

10 Conclusions

Mobile ad hoc networks exhibit new vulnerabilities to security attacks. As opposed to traditional networks, mobile ad hoc networks do not rely on any infrastructure and central authorities, they can be highly dynamic and mobile and operate over unreliable wireless media. When designing protocols for mobile ad hoc networks, special care has to be taken to include security mechanisms for the increased requirements in this environment. Security mechanisms for traditional networks can not be readily applied to ad hoc networks, because they often rely on infrastructures or are not scalable to a large distributed and dynamic environment, although some approaches can serve as a basis and only need to be modified. New ways of distributing trust can be implemented by introducing the notion of friends and making their collaboration pay off. This paper identifies the special requirements of mobile ad hoc network security and introduces a scheme to cope with them by retaliating for malicious behavior and warning affiliated nodes to avoid bad experiences. Nodes learn not only from their own experience, but also from observing the neighborhood and from the experience of their friends. Preliminary simulation results have shown that observable attacks on forwarding and routing can be thwarted by the suggested scheme of detection, alerting and reaction. Security is a major challenge for mobile ad hoc networks, because good citizenship can not be assumed in an open world. Depending on the extent to which the security issues are addressed, people might be reluctant to use mobile ad hoc networks.

References

[1] Ross Anderson and Frank Stajano. The resurrecting duckling. Lecture Notes in Computer Science, Springer-Verlag, 1999.

[2] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of IEEE Conference on Security and Privacy, Oakland, CA*, 1996.

[3] Levente Buttyan and Jean-Pierre Hubaux. Enforcing service availability in mobile ad-hoc wans. MobiHOC, 2000.

[4] Richard Dawkins. *The Selfish Gene.* Oxford University Press, 1989 edition, 1976.

[5] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the Sixth Annual ACM Symposium on Principles of distributed computing, Vancouver Canada*, pages 1 – 12, August 1987.

[6] Andreas Fasbender, Dogan Kesdogan, and Olaf Kubitz. Variable and scalable security: Protection of loccation information in mobile IP. In *Proceedings of the 46th IEEE Vehicular Technology Conference, Atlanta*, pages 963 – 967, 1996.

[7] Zygmunt Haas. Securing ad hoc networks. In *IEEE Network magazine, special issue on networking security, Vol. 13, No. 6, November/Dezember*, pages 24 – 30, 1999.

[8] http://www.sixdegrees.com.

[9] Jean-Pierre Hubaux, Jean-Yves Le Boudec, Silvia Giordano, and Maher Hamdi. The terminode project: Towards mobile ad hoc WANs. In *Proceedings of MOMUC'99 San Diego*, 1999.

[10] Mobile Ad Hoc Networks (MANET) Charter WG IETF. http://www.ietf.org/html.charters/manet-charter.html, 2000.

[11] Dave B. Johnson and David A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, Mobile Ad Hoc Network (MANET) Working Group, IETF, October 1999.

[12] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. Cornell Computer Science Technical Report 99-1776, 1999.

[13] Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of MOBICOM 2000*, pages 255 – 265, 2000.

[14] Radia Perlman. Network layer protocols with byzantine robustness. PhD. Thesis Massachussetts Institute of Technology, 1988.

[15] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45 – 48, 2000.

[16] Bruce Schneier. *Secrets and Lies. Digital Security in a Networked World.* John Wiley $ Sons, Inc, 1 edition, 2000.

[17] Bradley R. Smith, Shree Murthy, and J.J. Garcia-Luna-Aceves. Securing distance-vector routing protocols. In *Proceedings of Internet Society Symposium on Network and Distributed System Security, San Diego, CA*, pages 85 – 92, February 1997.

[18] William Stallings. *Network and Internetwork Security.* IEEE Press, 2 edition, 1995.

[19] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A library for parallel simulation of large-scale wireless networks. Proceedings of the 12th Workshop on Parallel and Distributed Simulations – PADS '98, May 26-29, in Banff, Alberta, Canada, 1998.

[20] P. Zimmerman. Pgp user's guide, 1993.